

## Verification of Translation

U.S. Patent Application No.: 09/785,994

Title of the Invention:

FACILITATOR HAVING A DISTRIBUTED CONFIGURATION,  
A DUAL CELL APPARATUS USED FOR THE SAME, AND  
AN INTEGRATED CELL APPARATUS USED FOR THE SAME

I, Yoko MATSUI, professional patent translator, whose full post office address is IKEUCHI & SATO Patent Office, Umeda Plaza Building, Suite 401, 3 – 25, Nishitenma 4-Chome, Kitaku, Osaka-shi, OSAKA 530-0047, Japan am the translator of the document attached and I state that the following is a true translation to the best of my knowledge and belief of JP 10(1998)-240530 A.

At Osaka, Japan  
DATED this April 25, 2001

Signature of the translator

  
\_\_\_\_\_  
Yoko MATSUI

# BEST AVAILABLE COPY

PARTIAL TRANSLATION OF JP 10(1998)-240530 A

Publication Date: September 11, 1998

Title of the Invention: OBJECT-ORIENTED PROCESSING SYSTEM

Patent Application Number: 9-38829

Filing Date: February 24, 1997

Inventors: Tetsuyasu YAMADA et al.

Applicant: NIPPON TELEGR & TELEPH CORP <NTT>

## [Claims]

[Claim 1] An object-oriented processing system for conducting computer processing by using object-oriented software, comprising: object groups each composed of a plurality of objects directly linked to each other; and interface means provided to each of the object groups, for conducting access control between an object of a certain object group and an object of another object group, wherein the objects in different object groups are indirectly linked through the interface means.

[Claim 2] An object-oriented processing system according to claim 1, wherein a plurality of the object groups are provided with a plurality of respectively distributed processor modules, and communication control means for controlling communication between the respective processor modules in each of the plurality of processor modules, and the interface means conducts access control between the respective objects in different processor modules through the communication control means.

[Claim 3] An object-oriented processing system according to claim 1, wherein a plurality of the object groups are provided with a plurality of respectively distributed processor modules, and communication control means for conducting communication control between the respective processor modules and connection control between the interface means in a certain processor module, in each of the plurality of processor modules, and the interface

means conducts access control between objects in each of the object groups of different processor modules and between objects in a certain processor module through the communication control means.

[Claim 4] An object-oriented processing system according to claim 3, wherein the communication control means includes registration means for previously registering processor modules having each object of an access target from the interface means in a certain processor module, and means for searching for the registration means to specify a processor module of the access target, accessing the processor module of the access target through the corresponding interface means in the certain processor module when the specified processor module is another processor module, and sending an access request to the communication control means of the another processor module when the specified processor module is another processor module.

[Claim 5] An object-oriented processing system according to any one of claims 2 to 4, comprising linking means for conducting compile from a source code of the object; based on an outside instruction of selecting either a single processor module structure or a multi processor module structure, selecting either one of each interface means for directly conducting the access control without using the communication control means in the single processor module structure or each interface means for conducting the access control through the communication means in the multi processor module structure; and linking the selected interface means to each of the compiled object group.

[Claim 6] An object-oriented processing system according to any one of claims 1 to 5, wherein the interface means is composed of jump means for receiving an access request from an object in a certain group with respect to an object in another group, and specifying a corresponding group to send the access request, and mapping means for receiving an access request from the jump means of another group with respect to an object in a certain group, and specifying a corresponding object in the certain group to send the access

request.

[Claim 7] An object-oriented processing system according to claim 6, wherein the jump means includes a table for previously associating each access request from each object in a certain group to an object in another group with a transmission target, and searches for and specifies the another group corresponding to the access request sent from the object in the certain group based on the table, and the mapping means includes a table for previously associating each of the access requests with each object in the certain group, and searches for and specifies an object in the certain group corresponding to the access request sent from the jump means of the another group based on the table.

[Claim 8] An object-oriented processing system according to any one of claims 1 to 7, wherein the object group is composed of a collection of a plurality of objects realizing a predetermined function.

[Claim 9] An object-oriented processing system according to any one of claims 1 to 8, wherein, the object group is composed of a plurality of object classes and instances derived from each of the plurality of object classes.

(Page 2, right column, line 44 – Page 3, right column, line 30)

[Detailed description of the Invention]

[0001]

[Field of the Invention] The present invention relates to a computer system using an object-oriented software technique. In particular, the present invention relates to an object-oriented processing system suitable for efficiently managing software.

[0002]

[Prior art] Conventionally, by using an object-oriented technique for software controlling a computer system, it is attempted to enhance the productivity

and reliability of software, etc. Furthermore, as a system for enhancing the performance of a computer system, there is a multiprocessor distributed processing system. In the multiprocessor distributed processing system, one intended processing is conducted by being distributed among a plurality of processors, whereby high speed and high reliability of processing can be obtained.

[0003] Figure 7 is a block diagram showing an exemplary configuration of a multiprocessor distributed processing system using a conventional object-oriented software technique. In Figure 7, reference numerals 711, 712 denote processor modules respectively including a processor, 721, 722 denote communication control mechanisms (represented by “NCU” in the figure) for linking the processor modules 711, 712, and 731 to 73J, 73K to 73N denote individual objects (represented by “OB” in the figure). Each object 731 to 73N is linked on an external reference name basis showing an access target such as a function in each object. Therefore, each object 731 to 73J and 73K to 73N require a link between those having a reference relationship, and in all the combinations.

[0004] Therefore, in the case of replacing an object when a program is changed, it is required to check all the related reference relationships and newly conduct linking so that there will not be an excess or a deficiency. Furthermore, as shown in Figure 7, when each object (731 to 73J, 73K to 73N) is disposed in a distributed environment using a plurality of processors, it is required to determine whether an access target is in the identical module or in another module, and to conduct linking, considering whether an access is an access to an object in another module through the communication control mechanisms 711, 712 or an access to an object in the identical module.

[0005] As described above, in a system using a conventional object-oriented software, linking is conducted on an external reference name basis such as a

function in an object. Therefore, in the case where hundreds of or more of objects are required, the number of links between objects is increased, a link state becomes complicated, which takes much time for replacing an object when a program is changed. In particular, in a multiprocessor distributed environment using a plurality of processors, determination of an access target becomes complicated, and the number of links between objects such as a link with respect to an inter-processor communication mechanism and a link with respect to an object in a processor is remarkably increased; as a result, a time for replacing an object when a program is changed is further prolonged.

[0006] Furthermore, it is desired that software controlling a computer system can use the same source code in both a distributed environment by a multiprocessor and a non-distributed environment by a single processor. However, the use of the same source code in both the single processor environment and the multiprocessor distributed environment is difficult because additional information showing a link target is different. A technique of newly conducting linking of objects in such a distributed system is also shown, for example, in the description in Figure 4 on page 132 of "Nikkei Computer 1990 Special Ed. New Wave, Object-orientation" (issued by Nikkei BP in 1989).

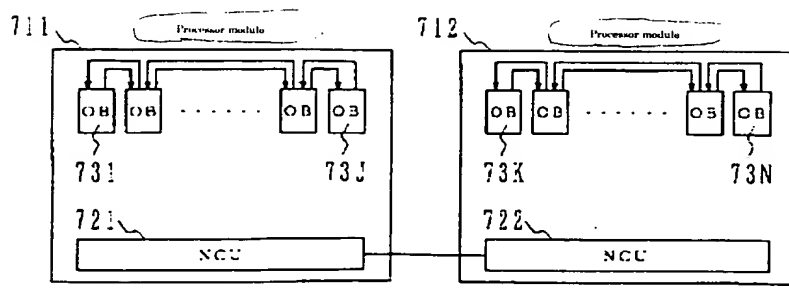
[0007]

[Problems to be Solved by the Invention] The problem to be solved is that a link is required in all the combinations of respective objects in the prior art. The object of the present invention is to overcome the problem of the prior art, and to provide an object-oriented processing system in which the number of links is decreased to simplify a link relationship between objects, replacement of an object in changing a program is made easy, the identical source code is applicable to both a distributed environment and a non-distributed environment, and such an application operation is made possible only by operating a tool.

[0008]

[Means for Solving the Problem] In order to achieve the above-mentioned object, according to the object-oriented processing system of the present invention, the respective objects constituting the system are grouped, and the respective objects in the same group are mutually directly link-connected on a reference name basis as in the prior art; however, link connection to the objects of the other group is indirectly performed through a specified object (interface part composed of a jumping processing part and a mapping processing part) provided in the respective groups. Thus, the respective objects of the different groups are connected only by access control between the specified objects provided in the respective groups and the entire link number is reduced.

Fig. 7





## PATENT ABSTRACTS OF JAPAN

(11) Publication number: **10240530 A**(43) Date of publication of application: **11.09.98**

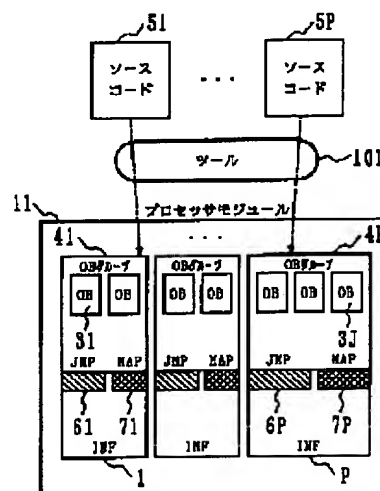
(51) Int. Cl.

**G06F 9/44**(21) Application number: **09038829**(22) Date of filing: **24.02.97**(71) Applicant: **NIPPON TELEGR & TELEPH  
CORP <NTT>**(72) Inventor: **YAMADA TETSUYASU  
SUNAGA HIROSHI****(54) OBJECT-ORIENTED PROCESSING SYSTEM****(57) Abstract:**

**PROBLEM TO BE SOLVED:** To provide an object-oriented processing system for reducing a link number, simplifying link relation between objects and easily replacing the object at the time of program change.

**SOLUTION:** The respective objects for constituting this system are grouped by a partial function unit for instance and the respective objects within a group are mutually directly link-connected for respective reference names as before, however, link connection to the objects of the other group is indirectly performed through a specified object (interface part 1 composed of a jump processing part 61 and a mapping processing part 71) provided in the respective groups. Thus, the respective objects of the different groups are connected only by access control between the specified objects provided in the respective groups and the entire link number is reduced.

COPYRIGHT: (C)1998,JPO



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-240530

(43) 公開日 平成10年(1998) 9月11日

(51) Int.Cl.<sup>6</sup>

G 0 6 F 9/44

識別記号

5 3 0

F I

G 0 6 F 9/44

5 3 0 C

審査請求 未請求 請求項の数9 O L (全 8 頁)

(21) 出願番号 特願平9-38829

(22) 出願日 平成9年(1997) 2月24日

(71) 出願人 000004226

日本電信電話株式会社

東京都新宿区西新宿三丁目19番2号

(72) 発明者 山田 哲靖

東京都新宿区西新宿三丁目19番2号 日本  
電信電話株式会社内

(72) 発明者 須永 宏

東京都新宿区西新宿三丁目19番2号 日本  
電信電話株式会社内

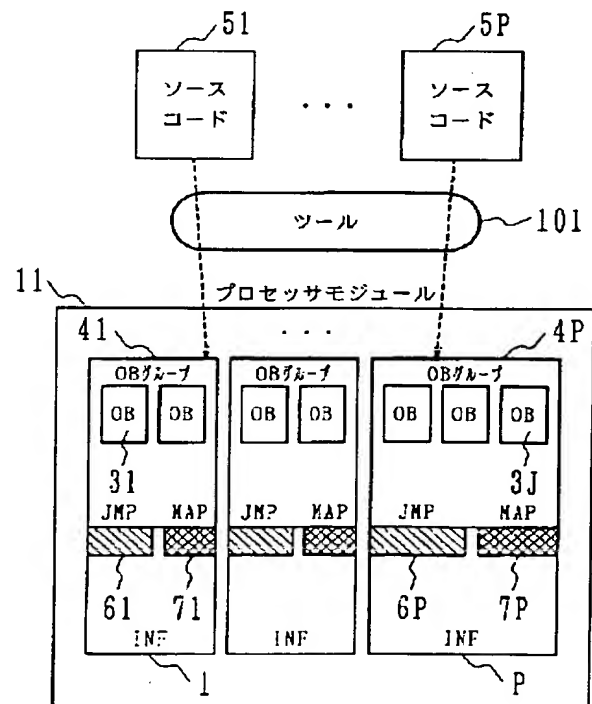
(74) 代理人 弁理士 磯村 雅俊 (外1名)

(54) 【発明の名称】 オブジェクト指向処理システム

(57) 【要約】

【課題】 各オブジェクトの全ての組合せにおいてリンクが必要な点。

【解決手段】 システムを構成する各オブジェクトを、例えば、部分的な機能単位でグループ化し、グループ内の各オブジェクト間には従来通り参照名毎に相互に直接リンク結合するが、他のグループのオブジェクトへのリンク結合は、各グループに設けられた特定のオブジェクト（ジャンプ処理部61とマッピング処理部71からなるインタフェース部1）を介して間接的に行う。このことにより、別グループの各オブジェクト間には、それぞれのグループに設けられた特定のオブジェクト間でのアクセス制御でのみ結合され、全体のリンク数を少なくすることができる。



## 【特許請求の範囲】

【請求項 1】 オブジェクト指向のソフトウェアを用いてコンピュータ処理を行うオブジェクト指向処理システムにおいて、相互に直接リンクされる複数のオブジェクトからなるオブジェクトグループと、該オブジェクトグループのそれぞれに、自オブジェクトグループにあるオブジェクトと他オブジェクトグループにあるオブジェクト間の相互のアクセス制御を行うインタフェース手段を設け、異なるオブジェクトグループにあるオブジェクト間を上記インタフェース手段を介して間接的にリンクすることを特徴とするオブジェクト指向処理システム。

【請求項 2】 請求項 1 に記載のオブジェクト指向処理システムにおいて、複数の上記オブジェクトグループがそれぞれ分散配置された複数のプロセッサモジュールと、該複数のプロセッサモジュールのそれぞれに、各プロセッサモジュール間での通信制御を行う通信制御手段を設け、上記インタフェース手段は、それぞれ異なるプロセッサモジュールにある上記各オブジェクト間の相互のアクセス制御を、上記通信制御手段を介して行うことを特徴とするオブジェクト指向処理システム。

【請求項 3】 請求項 1 に記載のオブジェクト指向処理システムにおいて、複数の上記オブジェクトグループがそれぞれ分散配置された複数のプロセッサモジュールと、該複数のプロセッサモジュールのそれぞれに、各プロセッサモジュール間での通信制御と自プロセッサモジュール内の上記インタフェース手段間の接続制御を行う通信制御手段を設け、上記インタフェース手段は、それぞれ異なるプロセッサモジュールの上記各オブジェクトグループのオブジェクト間および自プロセッサモジュール内のオブジェクト間の相互のアクセス制御を、上記通信制御手段を介して行うことを特徴とするオブジェクト指向処理システム。

【請求項 4】 請求項 3 に記載のオブジェクト指向処理システムにおいて、上記通信制御手段は、自プロセッサモジュール内の上記インタフェース手段からのアクセス先の各オブジェクトを有するプロセッサモジュールを予め登録した登録手段と、該登録手段を検索して上記アクセス先のプロセッサモジュールを特定し、該特定したプロセッサモジュールが他プロセッサモジュールであれば、自プロセッサモジュール内の当該するインタフェース手段を介してアクセスし、上記特定したプロセッサモジュールが他プロセッサモジュールであれば、該他プロセッサモジュールの上記通信制御手段にアクセス要求を送出する手段とを具備することを特徴とするオブジェクト指向処理システム。

【請求項 5】 請求項 2 から請求項 4 のいずれかに記載のオブジェクト指向処理システムにおいて、上記オブジェクトのソースコードからのコンパイルを行なうと共に、外部からの単一のプロセッサモジュール構成もしくは複数のプロセッサモジュール構成のいずれかの選択指

示に基づき、上記単一のプロセッサモジュール構成で上記通信制御手段を介さず上記アクセス制御を相互に直接行う各インタフェース手段、もしくは、上記複数のプロセッサモジュール構成で上記通信制御手段を介して上記アクセス制御を相互に行う各インタフェース手段のいずれか一方を選択し、該選択したインタフェース手段を上記コンパイルした各オブジェクトグループにリンクさせる結合手段を設けることを特徴とするオブジェクト指向処理システム。

【請求項 6】 請求項 1 から請求項 5 のいずれかに記載のオブジェクト指向処理システムにおいて、上記インタフェース手段は、自グループ内のオブジェクトからの他グループのオブジェクトへのアクセス要求を受け取り、対応するグループを特定して上記アクセス要求を送出するジャンプ手段と、自グループ内のオブジェクトへの他グループの上記ジャンプ手段からのアクセス要求を受け取り、自グループ内の対応するオブジェクトを特定して上記アクセス要求を送出するマッピング手段とからなることを特徴とするオブジェクト指向処理システム。

【請求項 7】 請求項 6 に記載のオブジェクト指向処理システムにおいて、上記ジャンプ手段は、上記自グループ内の各オブジェクトからの上記他グループのオブジェクトへの各アクセス要求の送出先を予め対応付けたテーブルを具備し、該テーブルに基づき、上記自グループ内のオブジェクトから送られてきたアクセス要求に対応する他グループを検索して特定し、上記マッピング手段は、上記アクセス要求のそれぞれを自グループ内の各オブジェクトに予め対応付けたテーブルを具備し、該テーブルに基づき、上記他グループのジャンプ手段から送られてきたアクセス要求に対応する自グループ内のオブジェクトを検索して特定することを特徴とするオブジェクト指向処理システム。

【請求項 8】 請求項 1 から請求項 7 のいずれかに記載のオブジェクト指向処理システムにおいて、上記オブジェクトグループは、所定の機能を実現する複数のオブジェクトの集合からなることを特徴とするオブジェクト指向処理システム。

【請求項 9】 請求項 1 から請求項 8 のいずれかに記載のオブジェクト指向処理システムにおいて、上記オブジェクトグループは、複数のオブジェクトクラスと、該複数のオブジェクトクラスのそれぞれから導出されたインスタンスからなることを特徴とするオブジェクト指向処理システム。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、オブジェクト指向のソフトウェア技術を用いたコンピュータシステムに係り、特に、ソフトウェアの管理を効率良く行うのに好適なオブジェクト指向処理システムに関するものである。

## 【0002】

【従来の技術】従来、コンピュータシステムを制御するソフトウェアにオブジェクト指向技術を用いることにより、ソフトウェアの生産性および信頼性の向上等が図られている。また、コンピュータシステムの性能を高めるものとして、マルチプロセッサ分散処理システムがある。このマルチプロセッサ分散処理システムでは、一つの目的とする処理を複数のプロセッサで分散して実行することにより、処理の高速化と高信頼化を図ることができる。

【0003】図7は、従来のオブジェクト指向のソフトウェア技術を用いたマルチプロセッサ分散処理システムの構成例を示すブロック図である。図7において、711、712はそれぞれ個別にプロセッサを含むプロセッサモジュール、721、722は両プロセッサモジュール711、712間を結合する通信制御機構（図中、「NCU」と記載）、731～73J、73K～73Nは個々のオブジェクト（図中、「OB」と記載）である。各オブジェクト731～73Nは、各オブジェクト内の関数などアクセス先を示す外部参照名毎にリンクが取られるので、各オブジェクト731～73J、73K～73Nは、参照関係を持つもの同士、全ての組合せにおいてリンクが必要である。

【0004】そのため、プログラム変更時にオブジェクトを差し替える場合、関連する参照関係を全てチェックして、過不足がないようにリンクを取り直す必要がある。さらに、本図7に示すように、各オブジェクト（731～73J、73K～73N）を複数プロセッサを用いた分散環境において配置すると、アクセス先が同一モジュールなのか、別のモジュールなのかを判断し、通信制御機構711、712を介しての別モジュールのオブジェクトへのアクセスか、同一モジュール内のオブジェクトへのアクセスなのかを踏まえてリンクする必要がある。

【0005】このように、従来のオブジェクト指向のソフトウェアを用いたシステムでは、オブジェクト内の関数など、外部参照名毎にリンクが取られるために、数百個以上のオブジェクトが必要な場合には、オブジェクト間リンクの数が増え、リンク状態が複雑になり、プログラム変更時におけるオブジェクトの差し替えに多くの時間を要してしまう。特に、複数プロセッサを用いたマルチプロセッサ分散環境においては、アクセス先の判断が複雑になり、また、プロセッサ間通信機構とのリンクやプロセッサ内オブジェクトとのリンクなど、オブジェクト間のリンク数が膨大になり、プログラム変更時におけるオブジェクト差し替え時間が、さらに増加してしまう。

【0006】また、コンピュータシステムを制御するソフトウェアは、マルチプロセッサによる分散環境でも、単一プロセッサによる非分散環境でも、同じソースコードを使えることが望まれるが、このように、同じソースコードを用いて、一プロセッサ環境およびマルチプロセ

ッサ分散環境の両方に適用することは、どこに対してリンクするかなどを示す付加情報が異なるので、困難であった。尚、このような分散システム上でのオブジェクトのリンクの取り直し技術に関しては、例えば、「日経コンピュータ 1990 特別編集版 新しい波、オブジェクト指向」（1989年、日経BP社発行）の第132頁の図4における説明でも示されている。

【0007】

【発明が解決しようとする課題】解決しようとする問題点は、従来の技術では、各オブジェクトの全ての組合せにおいてリンクが必要な点である。本発明の目的は、これら従来技術の課題を解決し、リンク数を減らしてオブジェクト間のリンク関係を簡素化し、プログラム変更時におけるオブジェクトの差替を容易とし、かつ、同一のソースコードを分散環境にも非分散環境にも適用でき、さらに、このような適用作業をツールの操作のみによって可能とするオブジェクト指向処理システムを提供することである。

【0008】

【課題を解決するための手段】上記目的を達成するため、本発明のオブジェクト指向処理システムは、システムを構成する各オブジェクトをグループ化し、同じグループ内の各オブジェクト間は従来通り参照名毎に相互に直接リンク結合するが、他のグループのオブジェクトへのリンク結合は、各グループに設けられた特定のオブジェクト（ジャンプ処理部とマッピング処理部からなるインタフェース部）を介して間接的に行うことにより、別グループの各オブジェクト間を、それぞれのグループに設けられた特定のオブジェクト間でのアクセス制御でのみ結合し、全体のリンク数を少なくする。

【0009】

【発明の実施の形態】以下、本発明の実施例を、図面により詳細に説明する。図1は、本発明のオブジェクト指向処理システムの本発明に係る構成の第1の実施例を示すブロック図である。本図1において、11はCPU（Central Processing Unit）を具備したプロセッサモジュール、31～3Jはプログラム部品としてのオブジェクト（図中、「OB」と記載）、41～4Pは複数のオブジェクトからなるオブジェクトグループ（図中、「OBグループ」と記載）、51～5Pは各オブジェクトグループ41～4Pに対応するソースコード、101はソースコード51～5Pをコンパイル等して各オブジェクトグループ41～4Pを生成するツール、1～Pは各オブジェクトグループ間のアクセス制御を行なうインタフェース部でありジャンプ処理部61～6Pとマッピング処理部71～7Pからなる。本例では、単一のプロセッサモジュール11内において、各オブジェクトグループ41～4P間で通信する例を示している。

【0010】各オブジェクトグループ41～4Pは、それぞれ、オブジェクト31～3Jを、例えば、部分機能

単位でグループ化したもので、各オブジェクト 3 1 ~ 3 J は、オブジェクトグループ 4 1 ~ 4 P のいずれかに含まれる。各インタフェース部 1 ~ P におけるジャンプ処理部 6 1 ~ 6 P は、プロセッサモジュール 1 1 内の自オブジェクトグループ 4 1 ~ 4 P の各オブジェクト 3 1 ~ 3 J から、プロセッサモジュール 1 1 内の他のオブジェクトグループ 4 1 ~ 4 P の各オブジェクト 3 1 ~ 3 J にアクセスするためのものであり、また、マッピング処理部 7 1 ~ 7 P は、プロセッサモジュール 1 1 内の他のオブジェクトグループ 4 1 ~ 4 P のジャンプ処理部 6 1 ~ 6 P からのアクセスを、自オブジェクトグループ 4 1 ~ 4 P 内の正当なオブジェクト 3 1 ~ 3 J のアドレスにマッピングするためのものであり、それぞれオブジェクトグループ 4 1 ~ 4 P 毎に設置されている。

【0011】このように、本実施例のオブジェクト指向処理システムにおいては、各オブジェクト 3 1 ~ 3 J を、部分的な機能単位でグループ化する。そして、同一のオブジェクトグループ 4 1 ~ 4 P 内においては、各オブジェクト 3 1 ~ 3 J 間には、従来通り、参照名毎に相互にリンク結合する。しかし、他のグループのオブジェクトへのアクセスは、各グループ毎に設けたジャンプ処理部 6 1 ~ 6 P およびマッピング処理部 7 1 ~ 7 P からなるインタフェース部 1 ~ P を介して行う。ジャンプ処理部 6 1 ~ 6 P は、図 2 (a) に示すテーブルを有し、各参照名に対応するグループ名を特定することができる。また、マッピング処理部 7 1 ~ 7 P は、図 2 (b) に示すテーブルを有し、各参照名に対応する実際のメモリ上のアドレスを特定することができる。

【0012】以下、このように構成されたオブジェクト指向処理システムの動作説明を、図 3 を用いて行なう。図 3 は、図 1 におけるオブジェクト指向処理システムの動作例を示すフローチャートである。本例は、図 1 におけるオブジェクト 3 1 からのアクセス要求に対する処理を示しており、まず、このアクセス要求が、同じオブジェクトグループ 4 1 内のオブジェクトに対すものかあるいは他のオブジェクトグループ内のオブジェクトに対すものか、すなわち、既に参照関係が特定されている所定のオブジェクトへのアクセス要求なのか否かを判別する (ステップ 301)。

【0013】所定のオブジェクトへのアクセス要求であれば、従来通り、その参照関係に基づき、所定のオブジェクトへアクセス要求を渡して実行し (ステップ 302)、所定のオブジェクトへのアクセス要求でなければ、アクセス要求を、同じグループ内のインタフェース部 1 に、参照名と共にを渡す (ステップ 303)。参照名とアクセス要求を受け取ったインタフェース部は、ジャンプ処理部 6 1 により、受け取った参照名をキーとして図 2 (a) のテーブルを検索し、オブジェクト 3 1 からのアクセス要求先のオブジェクトグループ (ここで、図 1 のオブジェクトグループ 4 P とする) を特定す

る (ステップ 304)。そして、この特定したオブジェクトグループ 4 P にアクセス要求と参照名を渡す (ステップ 305)。

【0014】オブジェクトグループ 4 P では、自グループ内のインタフェース部 P で、アクセス要求と参照名を受け取り、インタフェース部 P は、マッピング処理部 7 P により、参照名をキーとして図 2 (b) のテーブルを検索し、受け取った参照名に対応するメモリ上のアドレスを特定し (ステップ 306)、特定したアドレスからのメソッドをアクセス要求に対応して実行する (ステップ 307)。このように、本例では、別グループの各オブジェクト間は、それぞれのグループに設けられたインタフェース部間での間接ジャンプによるアクセス制御でのみ結合される。その結果、全体のリンク数を少なくすることができる。

【0015】次に、複数のプロセッサモジュールを用いた分散環境での本発明のオブジェクト指向処理システムの構成および動作を説明する。図 4 は、本発明のオブジェクト指向処理システムの本発明に係る構成の第 2 の実施例を示すブロック図である。本例では、各オブジェクトが二つのプロセッサモジュール 1 1 a、1 2 に分散配備された例を示しており、プロセッサモジュール 1 1 a は、図 1 のプロセッサモジュール 1 1 に、プロセッサモジュール 1 2 との通信制御を行う通信制御機構 (図中、「NCU」と記載) 2 1 と、この通信制御機構 2 1 で参照されるネームサーバ (図中、「Nサーバ」と記載) 1 1 1 を設け、さらに、図 1 のプロセッサモジュール 1 1 の各オブジェクトグループ 4 1 ~ 4 P におけるインタフェース部 1 ~ P のジャンプ処理部 6 1 ~ 6 P とマッピング処理部 7 1 ~ 7 P の代わりに、通信制御機構 2 1 との接続機能を有するジャンプ処理部 8 1 ~ 8 P とマッピング処理部 9 1 ~ 9 P を設けたものである。

【0016】プロセッサモジュール 1 1 a における各オブジェクトグループ (図中、「OBグループ」と記載) 4 1 a ~ 4 P a 内の各オブジェクト (図中、「OB」と記載) は、図 1 におけるプロセッサモジュール 1 1 におけるオブジェクト 3 1 ~ 3 J と同じものであり、ツール 101 によりソースコード 5 1 ~ 5 P から生成される。また、プロセッサモジュール 1 2 は、プロセッサモジュール 1 1 a と同様に、ツール 101 により各ソースコード 5 Q ~ 5 M から生成されたオブジェクト 3 K ~ 3 N とジャンプ処理部 8 Q ~ 8 M およびマッピング処理部 9 Q ~ 9 M からなるオブジェクトグループ 4 Q ~ 4 M と、通信制御機構 2 2、および、ネームサーバ 1 1 1 から構成されている。

【0017】各オブジェクトグループ 4 1 a ~ 4 P a、4 Q ~ 4 M は、それぞれ、他のプロセッサモジュール 1 1 a、1 2 の各オブジェクト 3 1 ~ 3 J、3 K ~ 3 N にアクセスする場合には、ジャンプ処理部 8 1 ~ 8 P、8 Q ~ 8 M を用いて通信制御機構 2 1、2 2 に結合し、ま

た、アクセスを受ける場合には、マッピング処理部 9 1 ~ 9 P、9 Q ~ 9 M を介して通信制御機構 2 1、2 2 に結合する。通信制御機構 2 1、2 2 は、後述の図 6 で示すように、ネームサーバ 1 1 1 に対する問合せ処理を行なうと共に、アルゴリズム処理部（図中、「ALG」と記載）1 2 1 を具備し、本発明に係る選択処理を行う。

【0018】ツール 1 0 1 は、各オブジェクトグループ 4 1 a ~ 4 P a、4 Q ~ 4 M 間のリンクをサポートするものであり、例えば、図 1 に示すような非分散環境に配備する時には、ジャンプ処理部 6 1 ~ 6 Q とマッピング処理部 7 1 ~ 7 Q を結合し、また、本例のように、各オブジェクトグループ 4 1 a ~ 4 P a、4 Q ~ 4 M に対応するソースコード 5 1 ~ 5 P、5 Q ~ 5 M を分散環境に配備する時には、ジャンプ処理部 8 1 ~ 8 P、8 Q ~ 8 M とマッピング処理部 9 1 ~ 9 P、9 Q ~ 9 M を、通信制御機構 2 1、2 2 を介して結合する。尚、本例のような分散環境においては、同じ種類のオブジェクトグループが各プロセッサモジュール 1 1 a、1 2 上に存在しても良い。

【0019】このように、本例のオブジェクト指向処理システムにおいては、ツール 1 0 1 は、分散環境、すなわち、マルチプロセッサ構成かどうかにより、相互に直接アクセス制御するインタフェース部（図 1 のジャンプ処理部 6 1 ~ 6 Q、マッピング処理部 7 1 ~ 7 Q）をリンクするか、通信制御機構 2 1、2 2 を介してアクセス制御するインタフェース部（図 4 のジャンプ処理部 8 1 ~ 8 P、8 Q ~ 8 M とマッピング処理部 9 1 ~ 9 P、9 Q ~ 9 M）をリンクするかの動作が異なる。以下、このようなツール 1 0 1 の動作説明を、図 5 を用いて説明する。

【0020】図 5 は、図 4 におけるツールの本発明に係る処理動作例を示す説明図である。ツール 1 0 1 は、まず、ソースコードをコンパイルして、各オブジェクトグループを生成し、操作者からの処理条件指定に従って、図 1 に示すように単一の同じプロセッサモジュール 1 1 に配備する場合は、各オブジェクトグループ 4 1 ~ 4 P に、ジャンプ処理部 6 1 ~ 6 P とマッピング処理部 7 1 ~ 7 P の処理プログラムを結合する。

【0021】また、操作者からの処理条件指定が、図 4 に示すように、各ソースコードを別々のプロセッサモジュール 1 1 a、1 2 に分散配置する場合は、オブジェクトグループ 4 1 a ~ 4 P a、4 Q ~ 4 M に、ジャンプ処理部 8 1 ~ 8 P、8 Q ~ 8 M とマッピング処理部 9 1 ~ 9 P、9 Q ~ 9 M の処理プログラムを結合する。さらに、これらジャンプ処理部 8 1 ~ 8 P、8 Q ~ 8 M とマッピング処理部 9 1 ~ 9 P、9 Q ~ 9 M を介して通信制御機構 2 1、2 2 を結合する。このようにしてマルチプロセッサ環境で結合されたオブジェクト指向処理システムの動作例を、図 6 を用いて説明する。

【0022】図 6 は、図 4 におけるオブジェクト指向処

理システムの動作例を示す説明図である。本例では、図 6 におけるオブジェクト 4 1 a からオブジェクト 4 M に通信する場合の通信制御機構 2 1 の動作を主に説明する。オブジェクト 4 1 a からジャンプ処理部 8 1 を通して、オブジェクト 4 M に対するリクエスト（アクセス要求）が通信制御機構 2 1 に送られる。通信制御機構 2 1 は、オブジェクト 4 M が、どこにあるかをネームサーバ 1 1 1 に問合せ。尚、この際、オブジェクト 4 M が複数個所に存在することもありうる。

【0023】オブジェクト 4 M の位置が分かると、通信制御機構 2 1 は、アルゴリズム処理部 1 2 1 により、まず、負荷分散等を考慮してオブジェクトを選択し、宛先がプログラム内部か、あるいは外部かを判定する。宛先がプロセッサ内部の場合は、当該するオブジェクトを起動する。また、宛先がプロセッサ外部の場合は、当該する他プロセッサモジュールの通信制御機構 2 2 と通信を行い、この通信制御機構 2 2 に、オブジェクト識別子と参照名を含んだリクエストを通知する。通信制御機構 2 2 は、マッピング処理部 9 M を介して、所定の参照名に対応するオブジェクトグループ 4 M 中のオブジェクトを起動する。

【0024】以上、図 1 ~ 図 6 を用いて説明したように、本実施例のオブジェクト指向処理システムでは、個々のオブジェクトを所定の部分機能単位でグループ化して、それぞれ異なるグループのオブジェクト間でのアクセス要求に関しては、ジャンプ処理部とマッピング処理部からなるインタフェース部を介して、グループ単位で結合する構成とする。また、マルチプロセッサの分散環境においては、インタフェース部と通信制御機構を介してアクセス制御を行う構成とする。このことにより、結合すべきリンク数が大幅に減り、リンケージが簡略化でき、変更等による差し替えが容易になる。また、非分散環境用のソースコードを、ツールを通すことによって、そのまま、マルチプロセッサの分散環境にも適用させることができ、さらに、ソフトウェアを分散環境で動作させるか非分散環境で動作させるかの切替を、ツールの操作により行うことができ、プログラムの負担を軽減できる。

【0025】尚、本発明は、図 1 ~ 図 6 を用いて説明した実施例に限定されるものではなく、その要旨を逸脱しない範囲において種々変更可能である。例えば、図 1 におけるジャンプ処理部 6 1 ~ 6 P が参照する図 2 (a) のテーブルでは、参照名とグループ名を対応付けているが、各オブジェクトグループ 4 1 ~ 4 P がアクセスすべき相手となる各オブジェクトグループ 4 1 ~ 4 P の参照名全てに対応するマッピング処理部 7 1 ~ 7 P のアドレスを登録する構成とすることでも良い。

【0026】また、本例では、マルチプロセッサの分散環境においては、異なるプロセッサモジュールにあるオブジェクト間のみならず、同じプロセッサモジュールに

ある異なるオブジェクトグループのオブジェクト間のアクセスに関しても、全て、通信制御機構を介して制御する構成としているが、異なるプロセッサモジュールにあるオブジェクト間のみを通信制御機構を介して制御し、同じプロセッサモジュールにある異なるオブジェクトグループのオブジェクト間のアクセスに関しては、インタフェース部間で直接制御する構成でも良い。この場合、図2 (a) におけるテーブルに参照名に対応して、グループ名と共にプロセッサモジュール名等を設けることで、アクセス制御の分岐を判別することができる。

# 【0027】

【発明の効果】本発明によれば、各オブジェクトの全ての組合せでのリンクが不要となり、リンク数を減らしてオブジェクト間のリンク関係を簡素化することができ、プログラム変更時におけるオブジェクトの差替が容易となり、かつ、同一のソースコードを分散環境にも非分散環境にも適用可能であり、さらに、このような適用作業をツールの操作のみによって行うことが可能である。

## 【図面の簡単な説明】

【図1】本発明のオブジェクト指向処理システムの本発明に係る構成の第1の実施例を示すブロック図である。

【図2】図1におけるジャンプ処理部が各参照名に対応するグループ名の特定に用いるテーブルおよびマッピング処理部が各参照名に対応する実際のメモリ上のアドレスの特定に用いるテーブルの構成例を示す説明図であ

る。

【図3】図1におけるオブジェクト指向処理システムの動作例を示すフローチャートである。

【図4】本発明のオブジェクト指向処理システムの本発明に係る構成の第2の実施例を示すブロック図である。

【図5】図4におけるツールの本発明に係る処理動作例を示す説明図である。

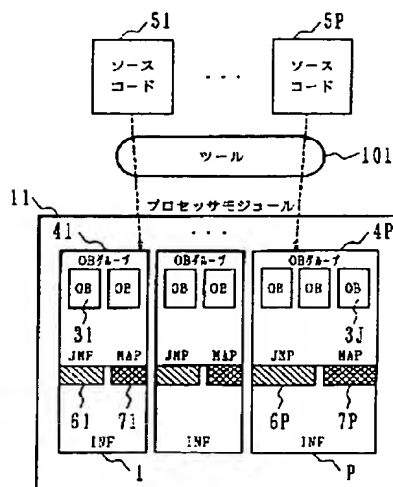
【図6】図4におけるオブジェクト指向処理システムの動作例を示す説明図である。

【図7】従来のオブジェクト指向のソフトウェア技術を用いたマルチプロセッサ分散処理システムの構成例を示すブロック図である。

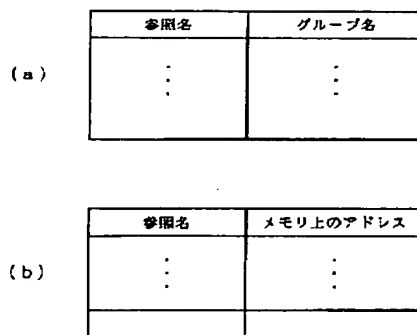
## 【符号の説明】

1～P：インタフェース部、11、11a、12：プロセッサモジュール、21、22：通信制御機構、31～3J、3K～3N：オブジェクト、41～4P、4Q～4M：オブジェクトグループ、51～5P、5Q～5M：ソースコード、61～6P：ジャンプ処理部、71～7P：マッピング処理部、81～8P、8Q～8M：ジャンプ処理部、91～9P、9Q～9M：マッピング処理部、101：ツール、111：ネームサーバ、121：アルゴリズム処理部、711、712：プロセッサモジュール、721、722：通信制御機構、731～73J、73K～73N：オブジェクト。

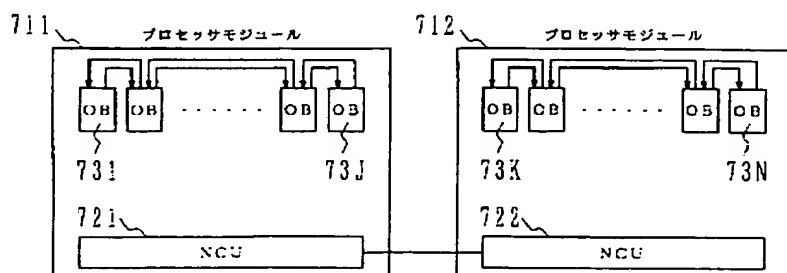
【図1】



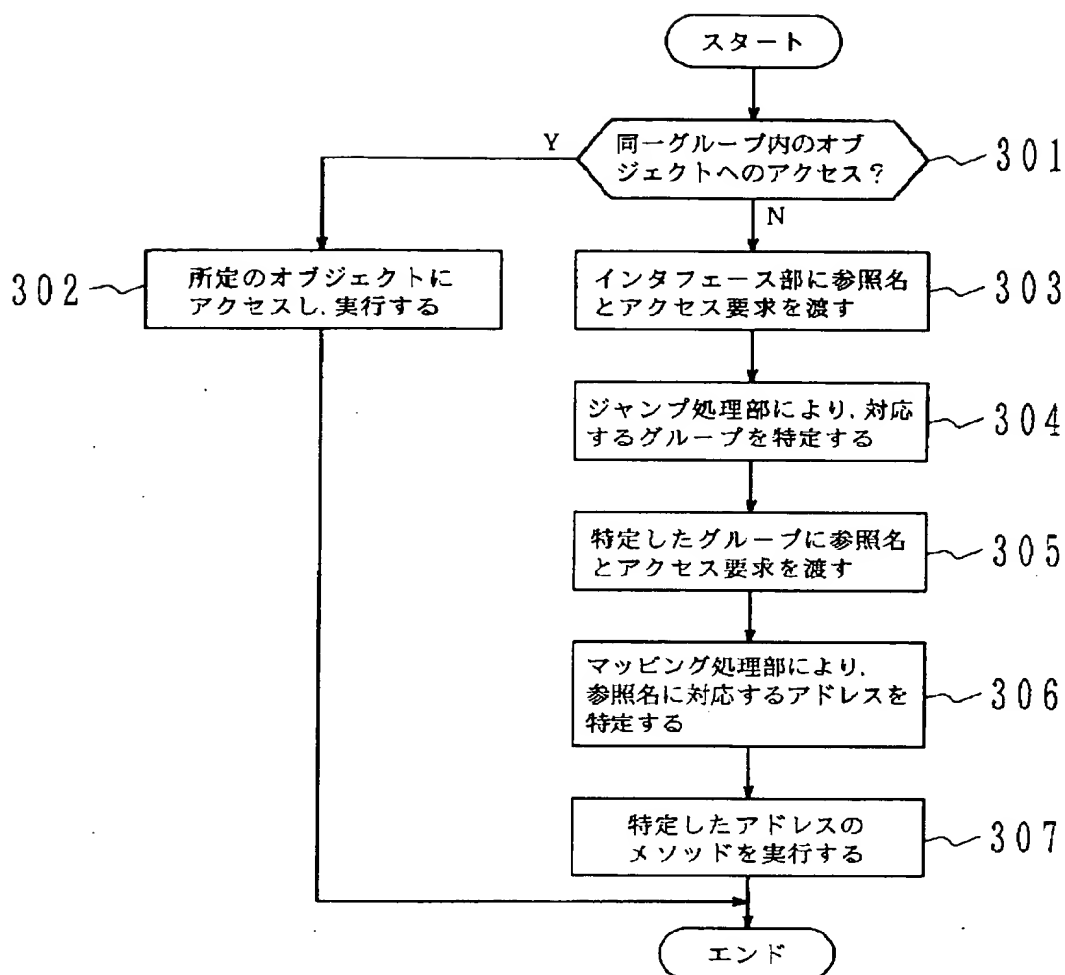
【図2】



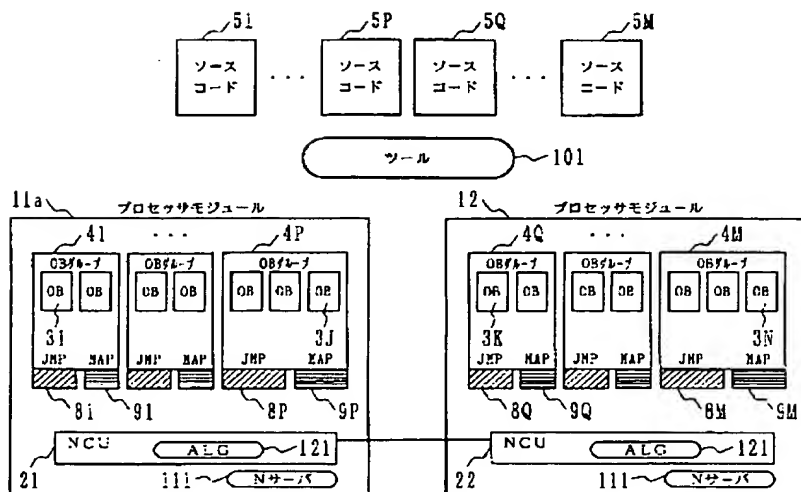
【図7】



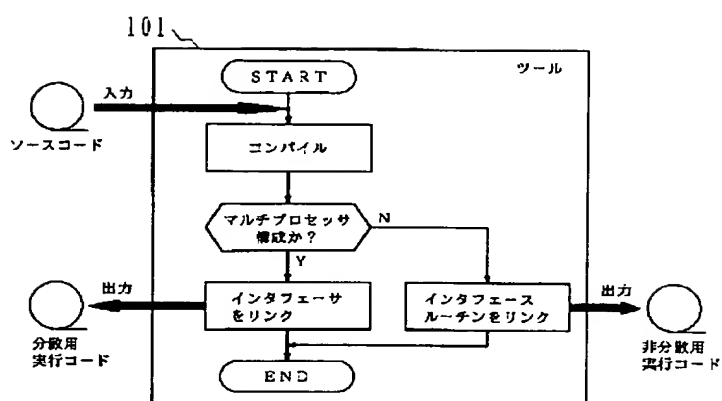
【図3】



【図4】



【図5】



【図6】

